

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1988	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE STARS Repository Guidebook Version 1.2			5. FUNDING NUMBERS MDA 903 84 C 0031 A-134	
6. AUTHOR(S) Catherine W. McDonald				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 1801 N. Beauregard St. Alexandria, VA 22311-1772			8. PERFORMING ORGANIZATION REPORT NUMBER IDA Memorandum Report M-385	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) STARS Joint Program Office 1400 Wilson Blvd. Arlington, VA 22209-2308			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, unlimited distribution; 06 September 1990.			12b. DISTRIBUTION CODE 2A	
13. ABSTRACT (Maximum 200 words) IDA Memorandum Report M-385, STARS Repository Guidebook Version 1.2, provides the Software Technology for Adaptable, Reliable Systems (STARS) Joint Program Office with a set of preliminary guidelines for developing and maintaining a software repository. The STARS Program plans to maintain several on-line, access-controlled software repositories for storing and distributing reusable software and related documentation. The STARS Program also plans research in the area of software management tools. These repositories are being established for the STARS community but may be accessed by other interested parties. The document discusses issues such as how software repositories should be established, how the code and documentation should be cataloged and retrieved, and other suggestions for operating effective and efficient software repositories.				
14. SUBJECT TERMS Software Repository; Software Management Tools; Software Documentation; Software Libraries; Information Storage and Retrieval Systems.			15. NUMBER OF PAGES 44	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

IDA MEMORANDUM REPORT M-385

STARS REPOSITORY GUIDEBOOK
Version 1.2

Catherine W. McDonald

April 1988

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
DARPA Assignment A-134



FOREWORD

The Institute for Defense Analyses (IDA) was asked by the STARS Joint Program Office (JPO) to look into the issues related to establishing one or more software repositories. This document provides a set of preliminary guidelines for developing and maintaining a software repository.

One area that is not sufficiently covered in this document is the development of an adequate taxonomy that will facilitate the search for and retrieval of reusable programs, packages, and generic software components. To date, repositories have proven to be so large and cumbersome that it is difficult to find anything useful in them. This document will look at ways to improve software repositories.

ACKNOWLEDGMENTS

The author wishes to thank a number of people who helped with this report. Members of the IDA Repository Planning Committee provided many helpful comments guidance in preparing this report. This committee included Terri Anderson, Reg Meeson, Katy Price, and Richard Waychoff. Jim Baldo, Bill Brykczynski, Nelson Corcoran, Clyde Roby and Robert Winner, also from IDA, spent many hours reviewing several drafts of this material and providing technical guidance. The author would also like to thank Col Joseph Greene of the STARS Joint Program Office for providing his insight into the preparation of a software repository.

UNCLASSIFIED

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	vii
1.0 INTRODUCTION	1
1.1 Purpose	1
1.2 Background	1
2.0 REPOSITORY REQUIREMENTS	2
2.1 Equipment Resources	2
2.2 Holdings	3
2.3 Compiler Selection	4
2.4 Standardization	4
3.0 APPROACH	6
3.1 Location of Repository	6
3.2 Submission of Data	7
3.3 Validation of Data	7
3.4 Access Control and Privileges	8
3.5 Accessing Data	8
4.0 INFORMATION STORAGE AND RETRIEVAL SYSTEMS	9
4.1 Automatic Indexing	10
4.1.1 Word Stem Generation Systems	10
4.1.2 Predefined Thesaurus Systems	11
4.1.3 Preset, Limited Vocabularies	12
4.2 Cataloging and Retrieval Systems	12
4.2.1 INFOTRAC	12
4.2.2 DIALOG	13
4.2.3 Hypertext	13
4.2.4 RUBRIC	14
4.2.5 Other Cataloging and Retrieval Systems	15
5.0 OTHER REPOSITORIES	17
5.1 National Research Laboratory	17
5.2 National Software Works	17
5.3 COSMIC	18
5.4 AdaNET	19
6.0 REFERENCES	20
APPENDIX I ACRONYMS	22
APPENDIX II DELIVERIES TO THE REPOSITORY	24

UNCLASSIFIED

1.0 INTRODUCTION

1.1 Purpose

The purpose of this IDA Memorandum Report is to provide the Software Technology for Adaptable, Reliable Systems (STARS) Joint Program Office with a set of preliminary guidelines for developing and maintaining a software repository. The STARS Program plans to maintain several on-line, access-controlled software repositories for storing and distributing reusable software and related documentation. The STARS office also plans to do research in the area of software management tools. These repositories are being established for the STARS community, but may be accessed by other interested parties.

The STARS Competing Prime Request for Proposal (RFP) [STARS 87] states that software deliveries for the STARS Program will not be complete until the code has been received by the STARS repository and compiled on a Department of Defense (DoD) validated Ada programming language compiler. The RFP also requires all software documentation to be in Standard Generalized Markup Language (SGML) [ISO 86] format and placed in the repository. These two requirements introduce new issues for the operation of a software repository operation.

Although there is some software library technology available today, this technology has been judged by the STARS Program Office to be inadequate for the needs of reusable software engineering technology, especially when the amount of code to be included in the repositories will exceed a few million lines of source code. This paper discusses the new issues outlined above, along with how new software repositories should be established, how the code and documentation should be catalogued and retrieved, and other suggestions for operating effective and efficient software repositories.

1.2 Background

The competing-prime concept was developed so that industry could better provide technology solutions for STARS. The task statements, progress reports, and incremental and final products of all participants will be shared through the repository mechanism. [STARS 86b].

The goal of the STARS program is to increase productivity while achieving greater system reliability and adaptability. This will be accomplished by providing integrated tools, reusable software components, and environments that are conducive to the development of reliable systems. One way to accomplish this goal is to provide a MILNET-accessible repository with access controls to support software reuse. In order to demonstrate and support reusability opportunities to reduce mission applications software costs, the STARS Repository will include a significant quantity of mission applications software that can be used to evaluate and advance software development approaches for reusable software. Each prime contractor should prepare a "Reusability Guideline" to be used by the software community when accessing software from the repository. Guidelines for the central STARS repository will be developed based on the prime contractors' guidelines.

2.0 REPOSITORY REQUIREMENTS

This section of the document discusses how the software repositories should be established. Specific areas of concern are the equipment necessary to operate the repositories in an efficient manner, what should constitute the holdings of the repositories, compiler selection and standardization within the repositories. The discussion outlines requirements found in the STARS RFP, STARS Program Management Plan and the STARS Technical Program Plan. Where lacking in direction, alternatives have been provided along with a suggested solution and the rationale for that solution.

2.1 Equipment Resources

The STARS *Technical Program Plan* states that a MILNET-accessible repository with access controls to support software reuse will be made available to the STARS community. The contractor selected to establish and maintain the repository must therefore be prepared to host the repository on the Defense Data Network (DDN).

The DDN, operated and controlled by the Defense Communications Agency (DCA), exists for the following reasons: [CONN 87]

- It provides a common, reliable, rugged, and secure communications path between organizations within the DoD, including all major DoD commands.
- It facilitates the sharing of resources between organizations on the Internet (which include many universities, national research laboratories, and commercial research centers).
- It facilitates communications between people at the organizations on the Internet.
- It provides a testbed for further development in computer networking.

STARS contractors will be provided an on-line mechanism for accessing STARS software and technical documentation, along with a hard-copy capability and computer-to-microfiche capability. All software included in the repository must be written in the Ada programming language. Use of non-Ada interim repository support tools will be permitted; however, all tools must be able to support Ada software. The fact that all repository support tools will eventually be written in Ada should be a driving factor in the design process.

The STARS RFP requires that the prime contractor meet certain requirements when establishing the repository. These include:

- That the contractor will have the capability to place files on floppy disks, optical compact disks, and tapes.
- The contractor will provide 24-hour access with network and dial-in communications for remote access.

UNCLASSIFIED

- Adequate disk capacity will ensure immediate access for 97% of requests, with others to be accessible within one hour from an archive.
- Mail will be supported for personnel involved in the STARS program.
- At a minimum, a single repository machine should be able to support 12 remote users simultaneously.

Another criterion which should be included in the design of the repository is the availability of repository dumps, including directory structures. For some users of the repository, it may be easier to obtain tapes of the contents of the repository. A system must then be provided to notify these users when changes are made to any programs, packages, documentation, etc. that they have requested. One suggestion is that the repository include an electronic bulletin board listing recent changes and updates to files. The bulletin board would be made available upon request to all interested parties.

In conformance with the STARS RFP, the host machine will have several Ada compilers. Support software written for the repository should be in Ada and such software would be considered a deliverable to the repository. This software, at a minimum, should include a user query facility to locate delivered code, a taxonomy for locating reusable capabilities, a publication system for printing formatted reports, and general software tools for manipulating and examining Ada programming language source code and documentation.

2.2 Holdings

All primes and their subcontractors will be required to submit products and data to the repository. Products will include prototype software, production software, demonstration programs, and documentation. It is recommended that all contractors establish and maintain configuration control for all products using ANSI/IEEE Standard 828-1983.[IEEE 83] This standard provides minimum requirements for the preparation of a Software Configuration Management (SCM) Plan and pertains to the entire life cycle of the software. It would take minimal effort on the part of the STARS office or its designated contractor to tailor this standard to the requirements of the STARS program. No software deliveries will be complete until code has been received by a STARS repository and compiled using a validated Ada compiler. The repository manager will actively seek out Ada language software from other domains and capabilities of interest for inclusion in the repository.

According to the RFP, source code will typically be submitted in a set of files with a command file to compile in the presence of already existing modules and with appropriate test procedures and test data. The repository will automatically perform the compilation along with whatever test runs are prescribed and upon successful completion will install the new source code.

The repository will host style, standards, metrics, and documentation tools through which the incoming source code may be passed. Abstracted output of such tools will be part of the source code documentation. The first delivery of source code may be an interface specification or an Ada Process Design Language (PDL). The earliest possible delivery of PDL and software is encouraged. In many subtasks, the design is to be delivered before production development. Rapid dissemination of capabilities, or an announcement of

capabilities to be developed together with an interface specification, will allow reuse, reduce unnecessary duplications and allow other organizations to plan for the use of the tool. Further deliveries and modifications to submitted code will be managed by the configuration control system established by the STARS JPO.

2.3 Compiler Selection

As previously stated, the STARS repository will include several validated Ada programming language compilers on various machines. In selecting these compilers, the STARS JPO may select one of the following options:

- The STARS JPO may arbitrarily select compilers for use in the repository,
- STARS may extend an invitation to compiler vendors for the donation of a compiler for use within the STARS repository,
- A vendor will provide a compiler based on a STARS study of compilers, or
- STARS will ultimately purchase compilers based on its study of compilers.

The first option is the most unlikely. The STARS JPO needs to establish guidelines for selecting the compilers that will be placed in the software repositories. The Ada programming language compilers must be validated and include a selected tool set. The repository tool set should include, at a minimum, a configuration manager, linker, debugger, and editor. The STARS JPO could extend an offer for any compiler vendor to donate a copy of their compiler to the repository, but it should be understood that these compilers are not to be considered the official repository compilers. If appropriate compilers are not donated to the repository, the STARS JPO should conduct a study of all available validated Ada programming language compilers and then purchase the necessary compilers for the repository based on the results of the study. With compiler technology continuously improving, there should be a steady stream of high quality compilers placed in the software repositories so that many different compilers are available at any time.

2.4 Standardization

A major problem in software development is the efficient and effective understanding of software by individuals other than the original author. One way to solve this problem is to apply standards to software products. This allows individuals who are familiar with parts of the software to become familiar with the total software package. Standardization is critical to the STARS repository because:

- Users other than the originators will access and retrieve items for the repository,
- Items in the repository will be incorporated into other software products, and
- The repository will require various standards in order to be efficiently developed and maintained.

UNCLASSIFIED

Two areas of concern within standardization are format and documentation. The STARS Program Management Plan (PMP) states that all documentation submitted to the STARS repository will use SGML. SGML is an internationally accepted standard for describing the technical structure of publications. SGML provides a means for delivering and storing publication text in the most easily maintained and updated form. When stored in electronic files, documents may be marked up using general markup methods or special electronic types of markup designed for processing by computer applications. Such markup designs include [COOMBS 87]:

- *Punctuational:* Punctuational markup consist of the use of a closed set of marks to provide primarily syntactic information about written utterances.
- *Presentation:* Authors mark up high level entities within a document to make the presentation clearer. Such markup includes horizontal and vertical spacing, folios, page breaks, enumeration of lists and notes, and a host of ad hoc symbols and devices.
- *Procedural:* Procedural markup consists of commands indicating how text should be formatted.
- *Descriptive:* Descriptive markup indicates what a text element is. A Generalized Markup Language (GML) is a descriptive language generally implemented on top of a clearly distinct, user-accessible procedural language. (SGML us is this category.)
- *Referential:* Referential markup refers to entities external to the document and is replaced by those entities during processing.
- *Metamarkup:* Metamarkup provides authors and support personnel with a facility for controlling the interpretation of markup and for extending the vocabulary of descriptive markup languages.

With respect to SGML, STARS will develop standard Document Type Declarations (DTDs) for all documents placed in the STARS repository. The DTDs will formalize the document markup by specifying which elements can occur in a document and in what order. It will also allow for the markup in documents to be validated according to the type definitions.

3.0 APPROACH

Topics of discussion for establishing the STARS repository will include identification of the repository location, individuals authorized to access the repository, who may submit data (code plus documentation) to the repository, and how that data will be accessed.

3.1 Location of Repository

In accordance with the STARS Competing Prime RFP [STARS 87], the STARS prime contractors will establish and maintain their own repositories. Although not all the information found within the individual repositories may be pertinent to the STARS repository, parts of these repositories will be included in the central repository when it is established.

Prime contractors will be asked to submit proposals for the establishment of a repository. The government may sponsor one or more repositories operated on computers owned or leased by the competing prime lead contractors or the government. The lead contractors must be prepared to access and, in at least one case, host their repository on the Defense Data Network (DDN) and public networks. Delivery of software will not be considered complete until the code has been verified using one of the repository's validated DoD Ada compilers. The confirmation of this compilation will be conducted by peer review. Members of the peer review will be designated by the prime contractor in charge of the repository and approved by the Director, STARS JPO.

If the repository contractor wants to connect data-processing equipment to the DDN, he must supply a data-network interface that complies with all DDN protocol specifications. There are two types of interface: a terminal-emulation processor (TEP) or a full service interface. The TEP emulates a virtual terminal to exchange information between a terminal and a host while the full-service interfaces allows different hosts to exchange information while providing terminal emulation. The architecture used by the prime contractor should consist of layered protocols that decompose the software and hardware into sets of independent modules. Since this modular approach will make upgrading less complex, adoption of the International Standards Organization (ISO) Open System Interconnect (OSI) protocol standard should be simple when the DoD gradually phases out older protocols. Modularity will also allow the prime contractor to quickly respond to changing requirements in today's networks. IDA Paper P-2041 provides an analysis of the effects of this transition.[BALDO 87] Their findings indicate that:

"The motivation for transition to the ISO OSI communication protocols is interoperability, standardized hardware and software, and therefore, lower development time and costs. There is a strong desire by the DoD to obtain interoperability between current and planned military and commercial communication networks. At present, OSI communication protocols are being developed for the commercial sector, which will begin to purchase such systems as soon as mature products become available.

NATO has also declared that all member countries will use ISO OSI communication protocols in their communication systems. The ability to utilize commercially available products that adhere to accepted international standards enables the DoD to benefit from using Commercial-Off-The-Shelf

UNCLASSIFIED

(COTS) hardware and software communication products, which will result in lower development time and costs."

3.2 Submission of Data

Data will be submitted to the repository from DoD agencies, DoD contractors, and members of the software community in general. All deliverables from prime contractors will be sent to the repository in electronic form.

The STARS repository is for all STARS deliverables: code, tools and administrative documents. No task under the STARS program will be considered complete until all items have been installed and validated by peer reviewers or repository personnel. The repository will also accept non-STARS products; however, these materials will not necessarily be endorsed by STARS.

Incremental deliveries may be made for both reports and code. Draft reports, starting with sections taken from the proposal, may be placed in the repository or offered in hard copy. Sections of documents should be configuration controlled so that partial deliveries or deliveries from several sources for a single document are feasible.[STARS 87]

3.3 Validation of Data

It is necessary for all incoming software, data, and documents to be thoroughly reviewed for relevancy, validity, accuracy and completeness prior to inclusion in the repository. Reviews should be conducted by computer professionals within the prime who have technical expertise in the area of software engineering. All deliverables to the software repositories will fall into one of the four categories described below:

- STARS-backed materials

Data which is submitted as a STARS deliverable will be tested and evaluated through a peer review process prior to submission to the repository as STARS-backed material. Test and sample outputs of code will be available for this material.

- Produced by STARS but still undergoing testing

This software may still have some bugs in it. Copies of any trouble reports will be provided and new users are requested to submit their trouble reports to the STARS JPO. Requestors will be notified when testing is complete and the final software is delivered.

- Non-STARS products

The STARS repository does not guarantee this software, however, it did pass the tests that were sent with it. Again, the repository would like to obtain copies of any trouble reports on this software.

- Non-STARS products

No guarantees and no tests. The STARS repository manager felt this software may be useful to the STARS community.

3.4 Access Control and Privileges

The STARS repository will be made available to the general public; however, since the repository is sponsored by the DoD, DoD agency and contractor requests will be given priority. Requestors should be allowed read access privileges to determine which, if any, files they would like to obtain. This will eliminate any possibility of altering or deleting material.

3.5 Accessing Data

Data may be accessed from the repository by contacting the software repositories using electronic mail, US mail, telephone, etc. Materials may be requested in the form of tape, microfiche, disk, or electronic mail. The contractor will provide the material and monitor the files. Monitoring the files will provide the repository manager with the necessary data to prepare certain reports. Such reports will include a list of the most popular files, which organizations are utilizing the information available within the STARS repository, and who should be providing the STARS JPO with software evaluations.

The method for obtaining material is:

- Review the multiple indexes within the repository either on-line or hard copy.
- Provide written or verbal request for material to the repository manager's office.
- Either repository manager or requestor completes "Repository Request Form" (which will be available on-line) stating which file(s) are requested, how these files will be used, and when the project will be completed. The requestor also guarantees that at the end of the project, an evaluation form will be sent to the repository manager's office evaluating all material (code, tools, etc.) provided by the repository.

The only fees associated with accessing material from the repositories will be reproduction fees. These fees will aid the repositories in becoming self-sustaining. According to the STARS RFP, materials will be available on tapes, microfiche, disks (floppy or compact), or hard copy and there will be no charge by the software repositories for network file transfers. The repository managers will also establish procedures for providing releasable software and documentation to the Defense Technical Information Center (DTIC) and the National Technical Information Service (NTIS) Federal Software Exchange.

4.0 INFORMATION STORAGE AND RETRIEVAL SYSTEMS

As the need for information has grown, so have new methods for indexing and storing information more efficiently. In the past, one of the difficulties associated with information retrieval has been the often necessary tradeoff between current of information and completeness of data. The importance of an item is often an individual choice and what is unimportant to one user may be crucial to another.

Specific documentation practices must be adhered to when setting up the repository so that users of the repository can determine the contents and usability of a module with minimal effort. At a minimum, five possible levels of documentation are recommended:

Module Abstract	This is a brief abstract of 1/2 to 1 page which gives a preliminary indication of whether or not a module may be useful; this is intended for someone who has a very large number of modules to review for reuse.
User's Documentation	This is a multiple page document which includes subprogram specs, data structures if appropriate, exceptions and descriptions of each of the above. This contains all the information normally required by a user of a package.
Maintenance Document	This is a more voluminous document usually written in DoD-STD-2167 format intended for someone who must maintain the code or modify the code for an application. Occasionally someone who just wants to use the package as is may consult this document for parameters such as CPU efficiency or memory-usage and other parameters normally not of interest to such an individual.
Design Rationale Document	This document provides the rationale for development of the software, the methodology used, etc. This document will be included with the software and other documentation in the repository.
Version Description Document	This document identifies the software and hardware being delivered, i.e. name of components, partitioning diagrams, and documentation. This document will be completed when the contractor delivers the software and document to the repository.

UNCLASSIFIED

The levels of documentation described in this document are intended to help users to obtain the amount of documentation they require. Too often the user has only two choices: a terse abstract or a 100-page maintenance document. The abstract generally does not contain sufficient information to allow the user to determine if the software is really useful. The maintenance document must then be consulted. The maintenance document provides too much irrelevant information. The ultimate result is that it takes too long to evaluate software for reuse and consequently, software is not reused.

The advent of computer based information storage and retrieval systems has greatly enhanced the ability of a retrieval system to both store and later locate pieces of information. Essentially all retrieval systems consist of three basic parts. The first part is a set of information items, usually documents, the second is a set of requests for specific information, and third, there is some mechanism which exists to determine which documents match which requests. Usually, the mechanism involves matching index terms, key words or phrases within the document to the same terms used in the request. In many cases, assignment of those index terms can now be done by the computer automatically, thus eliminating the huge manpower output required for manual indexing. However, automatic indexing also has its limitations, which will be discussed in the next section.

4.1 Automatic Indexing

Several methodologies exist for creating an automatic indexing system. The basic system involves analyzing the frequency with which certain words appear in a given document. A common assumption is that a word with a medium frequency would be a better indexing term than a word with a very low or very high frequency. The cutoff points between low, medium and high frequency of occurrence depend on the individual user's needs.

Automatic indexing is usually accomplished in the following manner. The abstracts or free text of the newly submitted documents are searched for all unique words. These words are compared to a 'stop list', which for the English language contains about 250 non-discriminatory words such as 'a', 'the', 'about', and other such words. All words in the document which are on the stop list are deleted from consideration as index terms. Of the remaining words, any words which occur only once in only one document are also eliminated. At this point, all plurals are made singular by removing the final 's' and identical word stems are combined. By now, about 50% of the original words have been eliminated, but for large document collections, the number of words still remaining may be too many. Therefore, a determination must now be made about which are high and which are low frequency words. In most cases, high frequency words are those which occur in over 25% of the documents included in the retrieval system, while low frequency words are those occurring in less than 5% of the documents. Any words with frequencies outside of this 5-25% range are eliminated from the indexing list. Those words which remain constitute the final indexing vocabulary.

4.1.1 Word Stem Generation Systems

In some automatic indexing systems, the depluralization step is expanded and includes a method for removing word suffixes, and occasionally prefixes, to reduce the possible index terms to their word stems. These stems will have a higher frequency of occurrence than any of the variant forms. Using the stems as index terms enhances recall, since a greater number of potentially relevant items will be retrieved than with any single form alone. However,

UNCLASSIFIED

because of some peculiarities in English, allowances must be made to prevent these exceptions from producing erroneous word stems. Usually, these allowances require the following:

- (1) A minimum word stem length must remain after the suffix is removed. Thus, 'sing' does not become 's' after the removal of the common suffix, '-ing'.
- (2) Either the suffix removal process must be applied recursively to remove multiple suffixes, or multiple suffixes as well as single ones must be listed in the suffix dictionary. 'Effectiveness' would then be properly reduced to 'effect', instead of 'effective' resulting from only one recursion.
- (3) Transformational rules should be included to recode word stems which have morphological changes. This includes removing a double consonant at the end of a word, or correcting certain consonant changes, such as 'relief' becoming 'relieving'.
- (4) Any additional context sensitive rules should be applied as needed. Therefore, the suffix '-allic' would not be removed from 'met', or 'ryst'. [SALTON 83]

4.1.2 Predefined Thesaurus Systems

Another indexing method uses a previously generated thesaurus to assign designated key words, as well as synonyms and related words to a specific document. Typically, a thesaurus is included to broaden low frequency words and to narrow high frequency words, providing additional applicable indexing terms. As in the previous case, this step would be added prior to removing the non-discriminating high and low frequency words.

Using a thesaurus, however, leads to additional organizational complications. Initially, a thesaurus must be generated, although the manner in which this is done is not really relevant, and can be manual, semiautomatic, or fully automatic, which leads to the next issue. The second issue involves deciding which terms need to be included in the thesaurus. Once the terms have been chosen, a reasonable grouping pattern must be determined. Often, the thesaurus will contain a group of low frequency words paired with synonyms which are of higher frequency, thus attempting to improve the recall function.

The words included in the final thesaurus should be carefully defined to cover the desired subject area. This is especially true for ambiguous words which have several unrelated meanings depending on context. Also, within the thesaurus classes, each synonym should have roughly the same frequency and thus approximately the same chance of being matched to a query. If this is not the case, low precision may result. Finally, the use of a thesaurus should not permit high frequency, non-discriminatory words to remain as index terms, even if size restrictions are not exceeded by their inclusion. At a minimum, these words should be assigned to separate classes of their own, since combining them with lower frequency terms also reduces precision.

4.1.3 Preset, Limited Vocabularies

One final possibility for indexing is the use of a predetermined set of index terms. This method ensures that only relevant terms are assigned to a particular document. However, a user must either know, or have access to, the list of allowable index terms. Suitable cross references should also be allowed and planned into the automatic indexing process. For most purposes, the use of such limited key words is not feasible or desirable, but for small, or very specialized collections, the preset vocabulary can enhance the speed of indexing as well as increasing recall and precision.

The use of a specialized vocabulary can be further enhanced by allowing the joint use of a thesaurus in the indexing process as explained in the previous section. Thus, the user would be able to find a specific item while remaining within the confines of a limited vocabulary. As mentioned above, sufficient cross references between subject areas and terms should be included for fixed word sets.

Another alternative to the fixed vocabulary is to use a set classification scheme, like the Association for Computing Machinery's *CR Classification Scheme*. [ACM 87] This indexing plan is organized in four levels: first level nodes of general terms, a second and third level with successively more specific descriptions, and finally, on the fourth level, subject descriptors which complete the classification. A fairly complete index is included with the ordering system, which helps the user to locate the proper classification for a particular topic. Since this scheme is fairly topic specific, with several predefined cross references, it is not as restrictive as a straight limited vocabulary system. It allows for some user flexibility, but yet, it still remains within the acceptable range of fixed vocabulary sets.

4.2 Cataloging and Retrieval Systems

Several cataloging and retrieval systems currently being used on a large scale are worth examining for their applicability to STARS. Each of them has desirable attributes that should be researched carefully. While some of the existing systems apply specifically to library cataloging, similar principles may be used to determine the proper system for STARS.

4.2.1 INFOTRAC

Developed completely in-house, INFOTRAC serves as the cataloging and retrieval system for the library at Rensselaer Polytechnic Institute (RPI) in Troy, New York. The system uses the SPIRES database system, developed at Stanford University, as the main programming language, and has been revised several times to add updated features.

INFOTRAC maintains several databases, including one each for books, professional journals, periodicals, music and reserved items, such as homework solutions or additional class readings, with a total of over one half million items in the open-shelf collection alone. The system also connects to over 5,600 other libraries and 200 other databases off campus. The main database source for bibliographic purposes is one called OCLC, located in Ohio. OCLC provides database support to nearly every large library in the country, and is considered almost as complete as the Library of Congress.

UNCLASSIFIED

The best way to understand how INFOTRAC works is to follow the system through the process of inputting a new book, from the ordering stage until it is checked out by a user. First, the database connects to OCLC, and a check is made to see if the book is on file. If it is, a request is made to include the information about the book in the next update. This update comes once a week, in a tape form which is readable by the mainframe at RPI. Once received, the tape is run through five in-house programs which extract the desired information. OCLC maintains up to 100 fields of information on each entry, but INFOTRAC only uses 15 fields for its purposes. The hierarchical database at Rensselaer allows for local changes to be made to the data from OCLC and while keeping several preset relationships between fields, only select fields are actually used as index terms. So while a search can be made by author or title, it cannot be made by call number, although the call number is included in the reference information at the end of the search. An additional feature recently added to INFOTRAC allows the user to see if the selected reference is on the shelf or if it has been checked out, thus eliminating unnecessary search time. [THORNTON 88]

4.2.2 DIALOG

A product of Lockheed Information Systems, of Palo Alto, California, DIALOG maintains access to more than 250 databases in a wide range of topics. The total number of accessible items available is in excess of 119 million, and is one of the most comprehensive online systems of its kind. [DIALOG 86]

DIALOG is based on an inverted file system, which is the most common file system type for commercial databases. An inverted file structure consists of a main file, and a related index file. The index file contains pointers to the locations in the main file where a particular item can be found. Thus, if the term 'information' is found in the title of a document as the fifth word, the identifier 'TI5' or something similar will be found in the index file next to the word 'information'. [SALTON 83] In this manner, only the index file must be updated or searched for the term location. If the term is located in more than one document, as is likely to be the case, a document identification number must be assigned, and used in conjunction with the term identifier.

DIALOG is organized much like OCLC, in that each record may have 100 or more separate fields. However, each database within DIALOG selects its own particular fields to use as indexes. Once these indexes are selected, DIALOG has several special features to enhance the search. The results of any single search are grouped together and assigned a set number. Then, sets can be combined by using Boolean operations, with multiple operations allowed, as well as parentheses to alter the order of operation. A search term may also be truncated on either the right or left in order to search for the stem and one or more variant forms. Another useful DIALOG feature is the ability to search for pairs of adjacent words, or for word pairs within a certain number of words from one another. Searches may be made in one specific field, or in multiple fields, depending on the needs of the user.

4.2.3 Hypertext

Recently, mechanisms have been developed which allow direct access to machine-supported references from one textual file to another. New interfaces are available to the user. These interfaces provide the user with the ability to interact directly with these files and to allow new relationships to develop. This activity falls under the general category of hypertext.[CONKLIN 87] Within hypertext, windows are associated with objects within a

UNCLASSIFIED

database and links are present between these objects. These links may be graphically depicted or serve as pointers within the database. This new method of retrieval should be investigated by the STARS JPO as a quick and flexible way to get an index started within the STARS repository. A list of features that would be somewhat ideal within a hypertext system are:

- The database is a network of textual nodes which can be thought of as a kind of hyperdocument.
- Windows on the screen correspond to nodes in the database on a one-to-one basis, and each has a name or title which is always displayed in the window. However, only a small number of nodes are ever "open" on the screen at the same time.
- Standard window system operations are supported: windows can be repositioned, resized, closed and put aside as small window icons. The position and size of a window or icon are cues to remembering the contents of the window.
- Windows can contain any number of link icons which represent pointers to other nodes in the database. The link icon contains a short textual field which suggests the contents of the node it points to. Clicking on a link icon with the mouse causes the system to find the referenced node and to immediately open a new window for it on the screen.
- The user can easily create new nodes and new links to new nodes or to existing nodes.
- The database can be browsed in three ways:
 - By following links and opening windows successively to examine their contents,
 - By searching the network (or part of it) for some string, keyword, or attribute value, and
 - By navigating around the hyperdocument using a browser that displays the network graphically.

4.2.4 RUBRIC

The contractor for the STARS repository must examine several methods for finding useful information in the repository. Another possible method is described by a system called **RULE-Based Retrieval of Information by Computer (RUBRIC)**. [MCCUNE 86] The attributes for RUBRIC include:

- Queries should be posed at the user's own conceptual level, using his or her vocabulary of concepts and without requiring complex programming.

UNCLASSIFIED

- The number of documents retrieved should depend upon the user's needs.
- A logical, understandable, and intuitive explanation of why each document was retrieved should be available.
- Users should be able to experiment easily with and revise queries, in order to handle changing interests or to correct previous system responses.
- Users should be able to store queries for future use and for sharing with other users.

4.2.5 Other Cataloging and Retrieval Systems

The Storage and Information Retrieval System (STAIRS) is available through IBM, and is similar in many respects to DIALOG. However, STAIRS does not provide the databases to be searched as DIALOG does. The main advantage of STAIRS is the addition of a database management system.

STAIRS uses the document abstracts or free text for searching purposes. Its operation is almost identical to DIALOG, with the exception of the actual terminology used to conduct the searches. A unique feature of STAIRS is the rank capability. This process ranks retrieved documents in order of importance based on one of several pre-specified algorithms. The rank feature may be especially valuable to a user with a large number of documents to review.

Unfortunately, STAIRS can be quite expensive to use, and storage requirements are large. The addition of database management to the initial system resulted in a large increase in storage space. Also, the user must have access to a large IBM system to use STAIRS. [SALTON 83]

The final system to be considered is the MEDLARS system operated by the National Library of Medicine. Its databases are concentrated primarily in the area of biomedicine. MEDLARS consists of three linked files, the index file, the postings file, and the data file. All the information related to a specific record is contained in the data file, including a unique identification number. Any search terms are included in the index file, with links to the specific document and field where the term can be found. This link consists of a two part number. The first part identifies the location in the postings file where the information about the term begins. The second part of the number gives the number of postings associated with the term. The postings file contains the document identification numbers where the term is found.

Although the majority of commands in MEDLARS are similar to DIALOG, one additional restriction is imposed. Parentheses are not allowed, and the Boolean hierarchy must be adhered to strictly. However, searches may be combined to allow the hierarchy to be overruled in some sense.

UNCLASSIFIED

Because MEDLARS was the first international online retrieval system of its kind, it is very well known. Unfortunately, it suffers from low recall and precision, but the lessons learned from its construction and subsequent use have provided valuable insight into the design of other systems. [SALTON 83]

UNCLASSIFIED

5.0 OTHER REPOSITORIES

Other repositories are available for use by the software community. The STARS repository will initially reside on SIMTEL20 at the White Sands Missile Range (WSMR) in New Mexico. The temporary STARS repository will be organized similar to the Ada Software Repository (ASR) which now resides on SIMTEL20. Mr. Richard Conn has been contracted to maintain this repository. The ASR is described in detail in Richard Conn's book, *The Ada Software Repository and the Defense Data Network: A Resource Handbook*..[CONN 87] Mr. Conn's provides additional information on such topics as how to use the DDN, available tools, and other facilities located on the network along with information on the other following repositories located on SIMTEL20:

CPM	for CP/M users
CPMUG	the CP/M Users Group
MSDOS	for MSDOS (IBM PC and compatible) users
PC-BLUE	the PC/BLEUE Users Group
SIGM	the Special Interest Group in CP/M
UNIX	for UNIX users
ZSYS	for ZCPR3 and Z System users
MISC	miscellaneous items, such as TOPS-20 and VAX VMS

5.1 Naval Research Laboratory

The Naval Research Laboratory (NRL) has established a temporary software repository for the foundation projects under the STARS program. The common Ada foundations include tools and parts from twelve different areas: operating systems, data base management systems, user interfaces, command language, graphics, text processing, network/communication, run-time support, planning and optimization (mission), reusability assistance, design-integration-test, and others. These areas are the foundations of the prototype environments. Once the STARS repository on SIMTEL20 is established, the foundation projects repository will be incorporated into the STARS repository.

5.2 National Software Works

Available on Arpanet from 1975-1981, the National Software Works was part of a research contract at the Rome Air Development Center. During the six years of its existence, NSW provided a distributed software tool environment to users of Arpanet. It ran primarily on the DEC 20, and IBM 360-190 series machines.

One of the unique capabilities of NSW was its maintenance of a software tool catalog under a 'Works Manager', a tool host with the local system, separate from any one operating system. The 'Works Manager' maintained in memory the location identity of each individual tool. Different instantiations of a tool were allowed, and all instantiations were independent of the host system. In addition, NSW had a transparent environment, with a common command language residing above any host languages. This enabled users from many different systems to use the tools available on NSW without having to translate languages.

UNCLASSIFIED

Although no longer operational, the research conducted during the development and use of the National Software Works project yielded valuable insight into the formation and organization of future repositories.

5.3 COSMIC

The Computer Software Management and Information Center (COSMIC) repository is operated under contract by the University of Georgia. COSMIC is part of the larger Technology Utilization Program sponsored by the National Aeronautics and Space Administration (NASA). "COSMIC's mission is to facilitate the distribution of computer software which has been developed by NASA or NASA contractors and which has significant potential secondary applications." [NASA 87] Much of the research and applications software developed under NASA sponsorship is available to the public through COSMIC.

While one specific programming language is not mandated for acceptance in COSMIC, a majority of the existing COSMIC programs are written in FORTRAN. There are, however, indications that in the future that emphasis may shift toward Ada, since that is now the standard for both the Department of Defense and NASA.

Submission of software to COSMIC is open to any programs which are of interest to COSMIC subscribers, including other government agencies, as well as business and educational institutions. As with other repositories, thorough documentation of all phases of the program ensures that the software can be utilized by additional users with minimal assistance from COSMIC. This documentation is then photocopied by COSMIC for distribution to requestors. Since the documentation is supplied separately from the code, the user must evaluate the potential applicability of the program based solely on the written instructions. Once the program is selected for use, these same instructions must serve as the user's guide.

COSMIC does provide some initial program screening, although this is limited to a two phase submittal process. In the first phase, the program is compiled and linked, to determine if the program is operationally complete. Any errors or missing routines are noted and an attempt is made to ascertain their cause, based on the documentation and knowledge of the original machine specifics. The results of this check are carefully documented before proceeding further. The second phase, evaluation, reexamines the outcome of the checkout step and attempts to reconcile any remaining discrepancies. If it is not possible to remedy the errors based on the available knowledge, a request for more information from the submitter is made. Once both these phases are satisfactorily completed, the package is accepted as part of the repository and becomes available for public use.

Once accepted, a program abstract is prepared by the COSMIC staff, along with keyword references based on NASA's thesaurus. Then, the package attributes are carefully documented and maintained as part of a master database. From here, various program characteristics, such as host environment requirements, distribution restrictions, number of lines of code, and other general areas of interest, can be searched and accessed. Records are also kept of who has acquired which programs and documentation, although this is not automatically integrated with the previously mentioned database. The programs are not tested other than for completeness, and COSMIC does not rate the programs in any way.[NASA 87]

5.3 AdaNET

Although it is not yet operational, the establishment of an AdaNET contract in October 1987, appears to be a very promising step toward the development of a viable Ada repository. MountainNet, the primary sponsor of AdaNET, is teaming with the Ada Joint Program Office (AJPO), the Office of Productivity, Technology, and Innovation (OPTI) within the Department of Commerce, NASA's Technology Utilization Division, several academic institutions, as well as several corporate sponsors, in order to define as many aspects of the repository as possible based on current knowledge. Any information determined as necessary for the repository that is not readily available through the sponsors will be researched, or developed, whichever is more effective. The overall goal for AdaNET is to provide an "advanced development network for Ada software applications", which will include centralized resources for Ada information and technology, continuous evaluation and development of new Ada tools and techniques, and consistent user support in the area of Ada applications as well as instruction and training in new or unfamiliar areas.[ADANET 87]

Most of the members of the project have experienced frustration when trying to access other repositories, and they plan to provide an alternative to the hassle through AdaNET. Several of these anticipated features are worth noting. First, a complete set of documentation must accompany any software submitted to AdaNET, which is not much different than any other repository. However, the documentation within AdaNET will likely follow Military-Standard 2167, or the newer version 2167A, which closely models the Ada life cycle. This requirement will ensure that the documentation will be more standardized and thus more beneficial to other users not familiar with the code. Second, a full battery of tests will be run on all software before it is accepted for dissemination through the repository. This evaluation process will be more thorough than most testing done within other repositories. Not only will the software be checked for completeness, but also for accuracy and ease of usability. Once checked, the software will be easily identified by some type of flag, as having passed the testing phase. Some programs, under special circumstances, may be admitted without passing all the tests. These pieces of code would be usable by others at their own risk, since AdaNET could not guarantee their operability. In addition to testing and screening submitted software, tutorials would be available to first time users of AdaNET generated software packages. Finally, even though a specific system for storage and retrieval has not been selected or developed as of yet, an emphasis is being placed on minimizing the use of disk input/output due to cost, space, and time limitations. AdaNET will most likely develop its own Information Storage and Retrieval System using a "complex system of taxonomies for the module descriptors." [RAUTNER 88]

Thus far, in its early planning and development stages, AdaNET appears to be the answer to many problems encountered on other software repositories. However, it is still too early to determine if the technology is available, now or in the near future, to accomplish all its goals. Once these areas are sufficiently addressed, then the true usefulness of the system to the user will be determined.

6.0 REFERENCES

Another memorandum report, M-408, has been prepared in conjunction with this document. The "Repository Bibliography" includes all references used in preparing this document in addition to other references which may be beneficial to the STARS Program Office and the prime contractor in establishing and maintaining the STARS repository. Key words generated for this bibliography were derived from the ACM Classification Scheme, January 1987. Only references specific to this document are included in this section.

- [ACM87] Association for Computing Machinery, "Additions from 1983 to 1987 Version of CR Classification Scheme," *ACM's Computing Reviews*, 28/1 (January 1987) 5-46.
- [ADANET 87] MountainNet, "AdaNET: The Advanced Development Network for Ada Software Applications," Dellslow, WV, October 1987.
- [BALDO 87] Baldo, James and David LeVan, *Effect of Transition from DoD to ISO OSI Communication Protocols*, IDA Paper P-2041, Institute for Defense Analyses, 1987.
- [CONKLIN 87] Conklin, Jeff, "Hypertext: An Introduction and Survey," *IEEE Computer*, Volume 20, Number 9, September 1987, pages 17-41.
- [CONN 87] Conn, Richard, *The Ada Software Repository and the Defense Data Network: A Resource Handbook*, New York Zoetrope, 1987.
- [COOMBS 87] Coombs, James H., Allen H. Renear, and Steven J. DeRose, "Markup Systems and The Future of Scholarly Text Processing," *Communications of the ACM*, volume 30, Number 11, November 1987, pp. 933-947.
- [DIALOG86] *1986 DIALOG Database Catalog*, DIALOG Information Services, Inc., March 1986.
- [DUVALL 76] Duvall, Lorraine W., *Software Data Repository Study*, IIT Research Institute, RADC-TR-76-387, December 1976.
- [GIBSON87] Telephone conversation between Terri Anderson, IDA and John Gibson, Director of COSMIC, December 22, 1987.
- [IEEE 83] IEEE Computer Society, ANSI/IEEE Standard 828-1983, *IEEE Standard for Software Configuration Management Plans*, New York, 1983.
- [ISO 86] International Standards Organization, ISO 8879-1986, *Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)*, 15 October 1986.

UNCLASSIFIED

- [JONES 85] Jones, Vill, Steve Litvintchouk, Jerry Mungle, Herb Krasner, John Melby, and Herb Willman, "Issues in Software Reusability." *ACM SIGSOFT Software Engineering Notes*, Volume 10, Number 2, April 1985, pp 108-109.
- [KUBAT 87] Kubat, Michael E., "Modularity is the Key to Interfacing to the Defense Date Network," *EDN*, 20 August 1987, pp 147-154.
- [MCCUNE 85] McCune, Brian, Richard M. Tong, Jeffrey S. Dean, and Daniel G. Shapiro, "RUBRIC: A System for Rule-Based Information Retrieval," *IEEE Transactions on Software Engineering*, SE-11/9, September 1985, pp 939-944.
- [METZGER87] Telephone conversation between Terri Anderson, IDA, and Dick Metzger, National Software Works, Rome Air Development Center, December 22, 1987.
- [NASA87] National Aeronautics and Space Administration, "COSMIC software Submittal Guidelines," Athens, GA, January 1987.
- [RAUTNER88] Telephone conversation between Terri Anderson, IDA, and James Rautner, Project Manager for AdaNET, January 4, 1988.
- [SALTON83] Salton, G. and Mr. McGill, *Introduction to Modern Information Retrieval*, New York: McGraw-Hill, 1983.
- [STARS 87] STARS JPO, *Software Technology for Adaptable, Reliable Systems (STARS) Competing Primes Contracts*, Washington, D.C., 1987.
- [STARS 86a] STARS JPO, *Software Technology for Adaptable, Reliable Systems (STARS) Program Management Plan*, Washington, D.C., 6 August 1986.
- [STARS 86b] STARS JPO, *Software Technology for Adaptable, Reliable Systems (STARS) Technical Program Plan*, Washington, D.C., 6 August 1986.
- [THORTON88] Telephone conversation with Joe Thornton, RPI Library, January 6, 1988.

UNCLASSIFIED

APPENDIX I

ACRONYMS

ANSI	American National Standards Institute
ASR	Ada Software Repository
COTS	Commercial-Off-The-Shelf
DCA	Defense Communications Agency
DDN	Defense Data Network
DoD	Department of Defense
DTD	Document Type Declaration
DTIC	Defense Technical Information Center
GML	Generalized Markup Language
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
JPL	Jet Propulsion Laboratory
JPO	Joint Program Office
NRL	Naval Research Laboratory
NSW	National Software Works
NTIS	National Technical Information Service
OSI	Open System Interconnection
PDL	Process Design Language
PMP	Program Management Plan
RADC	Rome Air Development Center

UNCLASSIFIED

RFP	Request for Proposal
RUBRIC	Rule-Based Retrieval of Information by Computer
SCM	Software Configuration Management
SGML	Standard Generalized Markup Language
SIMTEL20	SIMulation and TELeprocessing DECSYSTEM-20
SJPO	STARS Joint Program Office
STARS	Software Technology for Adaptable, Reliable Systems
TEP	Terminal-Emulation Processor
TPP	Technical Program Plan
WSMR	White Sands Missile Range

UNCLASSIFIED

APPENDIX II DELIVERIES TO THE REPOSITORY

This appendix provides a list of scheduled STARS deliverables (based on the RFP) that will be among the first items entered in the STARS repository. The name of the document is in the left column and the format it will be delivered in is in the right column. These documents will be made available to the STARS community.

- | | |
|---|---|
| 1. Analysis of STARS Technical Program Plan | Deliver electronically in SGML format |
| 2. Analysis of STARS Program Management Plan | Deliver electronically in SGML format |
| 3. Analysis of STARS Environment Requirements | Deliver electronically in SGML format |
| 4. STARS Technology Risk Analysis | Deliver electronically in SGML format |
| 5. Analysis of STARS Return on Investment | Deliver electronically in SGML format
Code for models to be delivered electronically |
| 6. STARS Five-year Strategy | Deliver electronically in SGML format |
| 7. Analysis of STARS Shadow Demo Projects | Deliver electronically in SGML format
Code for models to be delivered electronically |
| 8. Analysis of SDME Virtual Interfaces | Document to be delivered electronically in free format |
| 9. Specification of STARS Virtual Interfaces | Document to be delivered electronically in free format |
| 10. Implement STARS Virtual Interfaces | Document to be delivered electronically in free format . Source code in electronic form |
| 11. Design Distributed Virtual Interfaces | Document to be delivered electronically in free format. Source code in electronic form |
| 12. Implement Distributed Virtual Interfaces | Document to be delivered electronically in free format. Source code in electronic form |
| 13. Lessons Learned for STARS Environment | Document to be delivered electronically in free format |
| 14. Identification of Common Capabilities | Document to be delivered electronically in SGML format. Source code in electronic form |

UNCLASSIFIED

- | | |
|---|---|
| 15. Improved Common Capabilities | Source code to be delivered to STARS Repository in electronic form |
| 16. Revised Tools with New Capabilities | Source code to be delivered to STARS Repository in electronic form |
| 17. Guidelines-Foundation Capabilities | Document to be delivered electronically in SGML format. Source code in electronic form |
| 18. Taxonomy for Ada Capabilities | Document to be delivered electronically in free format to STARS Repository |
| 19. Tools Recommended for Development | Document to be delivered electronically in free format to STARS Repository |
| 20. Design of Ada Environment Tools | Source code to be delivered electronically to STARS Repository in electronic form |
| 21. Implementation of Environment Tools | Source code to be delivered electronically to STARS Repository in electronic form |
| 22. Plan for Software Engineering Tools | Document to be delivered electronically in SGML format. Source code in electronic form |
| 23. STARS Technical Guidelines and Standard | Document to be delivered electronically in SGML format. Source code in electronic form |
| 24. SGML Document Standards-Repository | Document to be delivered electronically to STARS Repository when it is established |
| 25. SGML Document Standards-Processor | Source code to be delivered to STARS Repository electronic form |
| 26. Repository Configuration Control Plan (IEEE-tailored) | Document to be delivered electronically in SGML format. Source code in electronic form |
| 27. Description of Enhanced Repository | Document to be delivered electronically in SGML format |
| 28. Source Code for Enhanced Repository | Source code to be delivered to STARS Repository electronic form for any tools or control system |
| 29. Software Interface Standards | Document to be delivered electronically in SGML format. Source code in electronic form |

UNCLASSIFIED

- | | |
|---|--|
| 30. Ada Design Processor and Support Tools | Document to be delivered electronically in SGML format. Source code in electronic form |
| 31. Specification-Design Support Tools | Source code to be delivered to STARS Repository in electronic form |
| 32. Software Technical Documentation | Document to be delivered electronically in SGML format. Source code in electronic form |
| 33. Software Documentation Tools Class 1 | Source code to be delivered to STARS Repository in electronic form for any tools or control system |
| 34. Software Documentation Tools Class 2 | Source code to be delivered to STARS Repository in electronic form for any tools or control system |
| 35. Documentation Plan of Action | Document to be delivered electronically in SGML format |
| 36. Plan for Access Controlled Configuration Management | Document to be delivered electronically in SGML format |
| 37. Trusted System Capability Plan | Document to be delivered electronically in free format
Source code in electronic form |
| 38. Proposed Tools-Building Trusted System | Document to be delivered electronically in free format
Source code in electronic form |
| 39. Initial Tools for Trusted Software | Source code to be delivered to STARS Repository in electronic form for any tools or control system |
| 40. Study-Generation of Operating System | Document to be delivered electronically in free format. Source code in electronic form |
| 41. Simple Operating System | Source code and documentation to be delivered to STARS Repository |
| 42. Plan for Trusted Operating System | Document to be delivered electronically in free format. Source code in electronic form |
| 43. Trusted Operating System or Control System | Source code and documentation to be delivered to STARS Repository in electronic form for any tools |
| 44. New Tools for Trusted Software | Source code to be delivered to STARS Repository in electronic form for any tools or control system. |
| 45. Restudy-Generation of Operating System | Document to be delivered electronically in free format to STARS Repository. Source code in electronic form |

UNCLASSIFIED

- | | |
|--|---|
| 46. Certification Example and Tools | Source code to be delivered to STARS Repository in electronic form for any tools or control system. |
| 47. Trusted Operating System | Source code and documentation to be delivered to STARS Repository in electronic form for any tools or c system. |
| 48. New Tools for Trusted Software | Source code to be delivered to STARS Repository in electronic form for any tools or control system. |
| 49. Restudy-Generation of Operating System | Document to be delivered electronically in free format
Source code in electronic form |
| 50. Certification Example and Tools | Source code to be delivered to STARS Repository in electronic form for any tools or control system. |
| 51. Security Architectures | Document to be delivered electronically in free format
Source code in electronic form |
| 52. Plan for First Research Brief | Document to be delivered electronically in free format
Source code in electronic form |
| 53. Risk Reduction Results | Source code and documentation to be delivered to STARS Repository in electronic form |
| 54. General Software Development | Document to be delivered electronically in SGML format |
| 55. Subcontracting Plan | Document to be delivered electronically in free format |
| 56. Reports of Peer Reviews | Document to be delivered electronically to STARS Repository on the day of the review by the host. |
| 57. Report of Oversight Group | Document to be delivered electronically to STARS Repository on the day of the review by the host. |
| 58. Plan for Risk Reduction Management | Document to be delivered electronically to STARS Repository on the day of the review by the host. |

Distribution List for IDA Memorandum Report M-385

NAME AND ADDRESS	NUMBER OF COPIES
-------------------------	-------------------------

Sponsor

Dr. John F. Kramer Program Manager STARS DARPA/ISTO 1400 Wilson Blvd. Arlington, VA 22209-2308	4
---	---

Other

Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
---	---

IDA

General W.Y. Smith, HQ	1
Ms. Ruth L. Greenstein, HQ	1
Mr. Philip L. Major, HQ	1
Dr. Robert E. Roberts, HQ	1
Ms. Catherine W. McDonald, CSED	1
Dr. Richard J. Ivanetich, CSED	1
IDA Control & Distribution Vault	2